

CLAIMS

1. In a Java computing environment, a method of generating optional attributes in a Java class file, said method comprising:
 - 5 receiving as input a Java runtime optimization;
 - generating one or more optional attributes based on said Java runtime optimization; and
 - writing said one or more optional attributes in an attribute table portion of a Java class file.
- 10 2. A method as recited claim 1, wherein said one or more optional attributes are implemented as the last attribute in said attribute table portion of said Java class file.
- 15 3. A method as recited claim 1, wherein said method further comprises:
 - generating computer program code that implements an application programming interface suitable for loading said one or more optional attributes.
- 20 4. A method as recited claim 3, wherein said application programming interface can be used to read said one or more optional attributes from said Java class file.
- 25 5. A method as recited claim 4, wherein said application programming interface includes functions that can be used to read first, last, and next optional attributes in said Java class file.

6. A method as recited claim 4, wherein said application programming interface includes a function suitable for finding an optional attribute in said Java class file.

5 7. A method as recited claim 1, wherein said Java runtime optimization is stored in a database.

8. A method as recited in claim 7, wherein said database is generated by a compiler extension or a software tool suitable for analyzing a Java application.

9. A method as recited in claim 7, wherein said database is stored in a runtime performance manager that can interact with software modules that generate and load said one or more optional attributes.

10. A method as recited in claim 7, wherein said method further comprises:
updating said database to reflect generation of said one or more optional attributes.

11. In a Java computing environment, a Java optional attribute generator suitable for generation of optional attributes in a Java class file, said Java optional attribute generator operating to:

receive as input a Java runtime optimization;
generate one or more optional attributes based on said Java runtime optimization; and
write said one or more optional attributes in an attribute table portion of a Java class file.

12. A Java optional attribute generator as recited in claim 11, wherein said Java optional attribute generator operates to generate computer program code that implements an application programming interface suitable for loading said one or more optional attributes.

5

13. A Java optional attribute generator as recited in claim 11, wherein an application programming interface can be used to read said one or more optional attributes from said Java class file.

10 14. A Java optional attribute generator as recited in claim 11, wherein said Java runtime optimization is stored in a database.

15 15. A Java optional attribute generator as recited in claim 11, wherein said database is generated by a compiler extension or a software tool suitable for analyzing a Java application.

16. A Java optional attribute generator as recited in claim 11, wherein said database is stored in a runtime performance manager that can interact with software modules that generate and load said one or more optional attributes.

20

17. A Java optional attribute generator as recited in claim 11, wherein said optional attribute generator operates to update said database to reflect generation of said one or more optional attributes.

25

18. A Java optional attribute generator as recited in claim 11, wherein said optional attribute generator operates to generate a description of an optional attribute.

19. A Java optional attribute generator as recited in claim 18, wherein said description is in XML format.

- 5 20. A computer readable medium including computer program code for generating optional attributes in a Java class file, said computer readable medium comprising:

computer program code for receiving as input a Java runtime optimization;

- 10 computer program code for generating one or more optional attributes based on said Java runtime optimization; and

computer program code for writing said one or more optional attributes in an attribute table portion of a Java class file.

- 15 21. A computer readable medium as recited in claim 20, wherein said method further comprises:

generating computer program code that implements an application programming interface suitable for loading said one or more optional attributes.

20

22. A computer readable medium as recited in claim 21, wherein said Java runtime optimization is stored in a database.

23. A computer readable medium as recited in claim 22, wherein said
25 database is generated by a compiler extension or a software tool suitable for analyzing a Java application.

24. A computer readable medium as recited in claim 22, wherein said database is stored in a runtime performance manager that can interact with software modules that generate and load said one or more optional attributes.

5

25. A computer readable medium as recited in claim 24, wherein said method further comprises:

updating said database to reflect generation of said one or more optional attributes.

(10)
(11)
(12)
(13)
(14)
(15)
(16)
(17)
(18)
(19)
(20)
(21)
(22)
(23)
(24)
(25)
(26)
(27)
(28)
(29)
(30)
(31)
(32)
(33)
(34)
(35)
(36)
(37)
(38)
(39)
(40)
(41)
(42)
(43)
(44)
(45)
(46)
(47)
(48)
(49)
(50)
(51)
(52)
(53)
(54)
(55)
(56)
(57)
(58)
(59)
(60)
(61)
(62)
(63)
(64)
(65)
(66)
(67)
(68)
(69)
(70)
(71)
(72)
(73)
(74)
(75)
(76)
(77)
(78)
(79)
(80)
(81)
(82)
(83)
(84)
(85)
(86)
(87)
(88)
(89)
(90)
(91)
(92)
(93)
(94)
(95)
(96)
(97)
(98)
(99)
(100)